## Remarks

The Applicant respectfully requests reconsideration of this application in view of the following remarks. In this response, claims 1 and 3-4 have been amended. No claims have been cancelled. And, thirteen new claims, i.e., claims 8-20, have been added. Hence, upon entry of this amendment, claims 1-20 are presented for examination.

## Claim Rejections – 35 U.S.C. § 102
### Nicholson

In the Office action, the Examiner rejected claims 1 and 4-7 under 35 U.S.C. §102(e) for allegedly being anticipated by US Patent No. 6,631,519 of Nicholson et al. (hereafter "**Nicholson**"). The undersigned respectfully disagrees with the Examiner's characterization of the teachings of Nicholson.

As presently understood by the undersigned, Nicholson generally relates to a system for automatically *generating database schema and associated interface methods* based on schema definition data 202 (see Abstract and Fig. 2). Nicholson describes a component generator 220 of schema generator 210 *generating computer program source code 230 that implements components that access and/or persist objects in a database* (see Abstract; col. 6, ll. 58-62; and Fig. 2). The source code 230 and interface definition 232 generated by the component generator 220 is compiled by compiler 240 to create the component objects 242 (see col. 6, ll. 63-64; and Fig. 2). Nicholson suggests one way of generating the interface definition file 232 is by using Interface Definition Language (IDL) (see col. 7, ll. 14-19). Notably, this use of a definition language that must be interpreted and the need for recompilation responsive to a change in the definition language is something embodiments of the present invention seek to avoid (see pg. 2, ll. 1-13).

Briefly and by way of background, various embodiments of the present invention involve configuring a network device based on meta information stored in one or more metadata files. The meta information defines objects representing components of the network device and includes descriptions of the objects, configuration information associated with each of the objects and information regarding relationships among the

objects. In one embodiment, the meta information is in the form of network component class descriptions. Based on the meta information, a network device may be configured by reading the meta information, converting the meta information into a meta runtime object model in which the objects are configured in accordance with the configuration information and loading the objects onto the network device.

Various methods are also described for managing metadata. For example, the metadata files may also be used to restore an original configuration to the network device or to update the network devices configuration responsive to a change to one or more of the metadata files. The metadata files may also be revised to reflect configuration changes applied to the network device by comparing the network devices' current runtime object model to that represented by the current state of the meta information and updating the one or more metadata files to reflect the differences.

At any rate, turning now to the specific language of the claims and with the above brief overview of Nicholson and embodiments of the present invention in mind, the undersigned now submits the following arguments pointing out significant differences between the invention as claimed by the Applicant and the teachings of Nicholson.

Regarding claim 1, as amended, Nicholson does not teach or reasonably suggest at least the recited steps of "creating a metadata file … defining objects representative of a ***configuration*** of components ***of a router***" (emphasis added) or "loading the objects onto the router." The undersigned has performed an electronic search of the text of Nicholson and found only a single instance of the word "router" at line 48 of col. 4 of Nicholson in the context of Nicholson describing the hardware and operating environment and the fact that computer 20 may operate in a networked environment and may be connected to one or more remote computers. It is respectfully submitted that this one reference to the word "router" by Nicholson is insufficient for the Examiner to draw the conclusion that "Nicholson teaches a computerized method for managing router metadata" as stated by the Examiner at pg. 2 of the Office action.

Meanwhile, the portions of Nichols (i.e., col. 6, ll. 49-57 and col. 7, ll. 9-13) recited by the Examiner to support his conclusion that Nicholson teaches a "metadata

file defining objects in a router" provide no such support. For the Examiner's convenience, the relied upon passages of Nicholson are reproduced below:

> In one embodiment of the invention, the SQL generator 218 generates a variant of SQL known as Transact-SQL (T-SQL). *T-SQL is a scripting language used to manipulate SQL compatible databases. T-SQL supports standard SQL operations, such as creating, modifying, and removing tables in a database.* T-SQL adds the ability to insert, update and remove columns within the database. In other words, T-SQL adds the ability to initialize and manipulate actual data in the database.
> (emphasis added) (col. 6, ll. 49-57)

> Source code 230 includes definitions of interface methods that provide accessor [sic] methods to the properties and objects defined in the schema. In addition, the *source code includes interfaces to methods that manipulate objects in the database* that are persistent versions of the schema objects defined in the schema definition data. For example, *methods are defined in the source code 230 that read, write, and modify rows in tables created via the SQL generator 218*. Appendix C illustrates an exemplary source code ".h" file produced using the schema definition shown in Appendix A.
> (emphasis added) (col. 7, ll. 4-13)

The undersigned can find no reasonable interpretation of these paragraphs to support the Examiner's position that it teaches a "*metadata file defining objects in a router*." In fact, the portions of Nicholson support the undersigned's earlier conclusion that Nicholson is focused on automatically *generating database schema and associated interface methods* by *generating computer program source code for accessing and/or persist objects in a database*.

In the Office action, the Examiner indicates "loading the objects onto the router" is "implicit in Nicholson in order to route or switch the objects to other destination [sic]" citing col. 7, ll. 23-31. However, Nicholson does not purport to "route or switch" objects to other destinations as it is concerned only with persisting and accessing objects in a database (as defined by tables 250, stored procedures 242 and views 254). For at least these reasons, claim 1 and its dependent claims, which add

further limitations, are thought to be clearly distinguishable over the teachings of Nicholson.

Regarding dependent claim 4, it further limits the method of claim 1 by including "*creating a hash table of attribute names and attribute values* from the metadata file." Without explanation, the Examiner indicates "Nicholson teaches a hash table" in col. 10, 11 and 15. While the undersigned readily acknowledges the XML data shown in col. 10 is in a table format and includes attributes and values, it is certainly not a "hash table." A hash table is a data structure that divides all elements into categories or buckets, to allow quick access to the elements. A hash function determines which bucket an element belongs in. The XML data shown in col. 10 is not a table to/from which information is stored/retrieved based on a hash function. Meanwhile, the "tables" in col. 11 and 15 appear to be representative of auto-generated T-SQL script (see top of col. 11). Consequently, none of the portions of Nicholson relied upon to reject claim 4 teach or reasonably suggest a "hash table" as recited. For at least this additional reason, dependent claim 4 is thought to be further distinguishable over the teachings of Nicholson.

Regarding claim 6, the Examiner cites col. 2, ll. 5-11 in support of his conclusion that "Nicholson teaches comparing (matching) by the router the objects of the object model to a runtime object model." The undersigned sees the word "match" in this portion of Nicholson, but respectfully suggests the Examiner is reading more into the relied upon portion of Nicholson than is actually present. This portion of Nicholson uses the term "match" as "compatible with." Therefore, Nicholson generates interface definitions and provides for instantiating objects in a run-time environment that are compatible with objects in the database. There is no suggestion that any "comparison" occur. For at least this additional reason, dependent claim 6 is thought to be further distinguishable over the teachings of Nicholson.

## Claim Rejections – 35 U.S.C. § 103
### Nicholson in view of Schoening and Menzies

In the Office action, the Examiner rejected claim 2 under 35 U.S.C. 103(a) for allegedly being unpatentable over Nicholson in view of US Patent No. 6,226,788 of Schoening et al. (hereafter "**Schoening**") and US Patent No. 6,317,748 of Menzies et al. (hereafter "**Menzies**"). The undersigned respectfully disagrees with the Examiner's characterization of the individual and combined teachings of Nicholson, Schoening, and Menzies. Notably, the Examiner relies upon Schoening and Menzies only for their alleged teachings regarding the use of a SNMP MIB. However, Nicholson has been shown to be deficient with respect to other limitations not thought to be addressed by Schoening or Menzies. Consequently, for at least the reasons presented above with reference to claim 1, claim 2 is allowable over the relied upon combination.

### New Claims

Thirteen new claims, i.e., claims 8-20, have been added. No new matter has been introduced by these new claims.

One new independent claim, i.e., claim 8, has been added. The newly added independent claim is thought to be allowable over the references of record for at least various of the reasons presented above with reference to claim 1. For example, claim 8 requires "defining a plurality of objects that represent components of a network device by providing, in one or more metadata files … , meta information regarding the plurality of objects, including descriptions of the plurality of objects, configuration information associated with each of the plurality of objects and information regarding relationships among the plurality of objects in the form of network component class descriptions" and "configuring a network device based on the meta information by reading the meta information, converting the meta information into a meta runtime object model including the plurality of objects configured in accordance with the configuration information and loading the plurality of objects onto the network device."

## Conclusion

The undersigned respectfully submits that the remarks have overcome the rejections, and that the pending claims are in condition for allowance. Accordingly, the undersigned respectfully requests that the rejections be withdrawn and that a Notice of Allowance be promptly issued for claims 1-20.

## Request for a Telephone Interview

If the Examiner believes a telephone conference would expedite prosecution of this application, please telephone the undersigned at 303-284-5103.

Respectfully submitted,
HAMILTON & DESANCTIS

Michael A. DeSanctis, Esq.
Reg. No. 39,957
Customer No. 64128